

CSE 2001:
Introduction to Theory of Computation
Fall 2012

Suprakash Datta

datta@cse.yorku.ca

Office: CSEB 3043

Phone: 416-736-2100 ext 77875

Course page: <http://www.cs.yorku.ca/course/2001>

Next

- **Chapter 2:**
 - **Context-Free Languages (CFL)**
 - **Context-Free Grammars (CFG)**
 - **Chomsky Normal Form of CFG**
 - **$RL \subset CFL$**

Context-Free Languages (Ch. 2)

Context-free languages (CFLs) are a more powerful (augmented) model than FA.

CFLs allow us to describe non-regular languages like $\{ 0^n 1^n \mid n \geq 0 \}$

General idea: CFLs are languages that can be recognized by automata that have one single stack:

$\{ 0^n 1^n \mid n \geq 0 \}$ is a CFL

$\{ 0^n 1^n 0^n \mid n \geq 0 \}$ is not a CFL

Context-Free Grammars

Grammars: define/specify a language

Which simple machine produces the non-regular language $\{ 0^n 1^n \mid n \in \mathbb{N} \}$?

Start symbol S with rewrite rules:

- 1) $S \rightarrow 0S1$
- 2) $S \rightarrow \text{"stop"}$

S yields $0^n 1^n$ according to

$S \rightarrow 0S1 \rightarrow 00S11 \rightarrow \dots \rightarrow 0^n S 1^n \rightarrow 0^n 1^n$

Context-Free Grammars (Def.)

A context free grammar $G=(V,\Sigma,R,S)$ is defined by

- V : a finite set variables
- Σ : finite set terminals (with $V \cap \Sigma = \emptyset$)
- R : finite set of substitution rules $V \rightarrow (V \cup \Sigma)^*$
- S : start symbol $\in V$

The language of grammar G is denoted by $L(G)$:

$$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$$

Derivation $\Rightarrow^*$

A single step derivation “ $\Rightarrow$ ” consist of the substitution of a variable by a string according to a substitution rule.

Example: with the rule “ $A \rightarrow BB$ ”, we can have the derivation “ $01AB0 \Rightarrow 01BBBB0$ ”.

A sequence of several derivations (or none) is indicated by “ $\Rightarrow^*$ ”

Same example: “ $0AA \Rightarrow^* 0BBBBB$ ”

Some Remarks

The language $L(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$ contains only strings of terminals, not variables.

Notation: we summarize several rules, like

$A \rightarrow B$

$A \rightarrow 01$ by $A \rightarrow B \mid 01 \mid AA$

$A \rightarrow AA$

Unless stated otherwise: topmost rule concerns the start variable

Context-Free Grammars (Ex.)

Consider the CFG $G=(V,\Sigma,R,S)$ with

$$V = \{S\}$$

$$\Sigma = \{0,1\}$$

$$R: S \rightarrow 0S1 \mid 0Z1$$

$$Z \rightarrow 0Z \mid \varepsilon$$

Then $L(G) = \{0^i1^j \mid i \geq j\}$

S yields $0^{j+k}1^j$ according to:

$$\begin{aligned} S &\Rightarrow 0S1 \Rightarrow \dots \Rightarrow 0^jS1^j \Rightarrow 0^jZ1^j \Rightarrow 0^j0Z1^j \Rightarrow \\ &\dots \Rightarrow 0^{j+k}Z1^j \Rightarrow 0^{j+k}\varepsilon1^j = 0^{j+k}1^j \end{aligned}$$

Importance of CFL

Model for natural languages (Noam Chomsky)

Specification of programming languages:
“parsing of a computer program”

Describes mathematical structures

Intermediate between regular languages and
computable languages (Chapters 3,4,5 and 6)

Example Boolean Algebra

Consider the CFG $G=(V,\Sigma,R,S)$ with

$$V = \{S,Z\}$$

$$\Sigma = \{0,1,(,),\neg,\vee,\wedge\}$$

$$R: S \rightarrow 0 \mid 1 \mid \neg(S) \mid (S)\vee(S) \mid (S)\wedge(S)$$

Some elements of $L(G)$:

$$0$$

$$\neg((\neg(0))\vee(1))$$

$$(1)\vee((0)\wedge(0))$$

Note: Parentheses prevent “ $1\vee 0\wedge 0$ ” confusion.

Human Languages

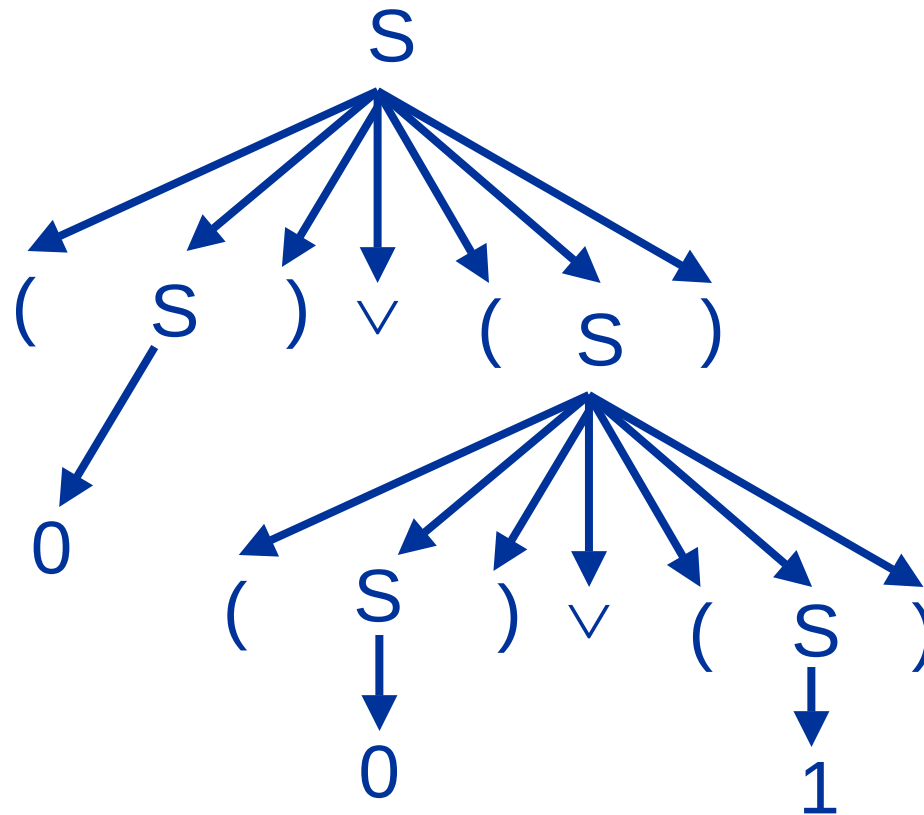
Number of rules:

<SENTENCE> → <NOUN-PHRASE> <VERB-PHRASE>
<NOUN-PHRASE> → <CMPLX-NOUN> | <CMPLX-NOUN> <PREP-PHRASE>
<VERB-PHRASE> → <CMPLX-VERB> | <CMPLX-VERB> <PREP-PHRASE>
<CMPLX-NOUN> → <ARTICLE> <NOUN>
<CMPLX-VERB> → <VERB> | <VERB> <NOUN-PHRASE> ...
<ARTICLE> → a | the
<NOUN> → boy | girl | house
<VERB> → sees | ignores

Possible element: the boy sees the girl

Parse Trees

The parse tree of $(0) \vee ((0) \wedge (1))$ via rule
 $S \rightarrow 0 \mid 1 \mid \neg(S) \mid (S) \vee (S) \mid (S) \wedge (S)$:



Ambiguity

A grammar is ambiguous if some strings are derived ambiguously.

A string is derived ambiguously if it has more than one leftmost derivations.

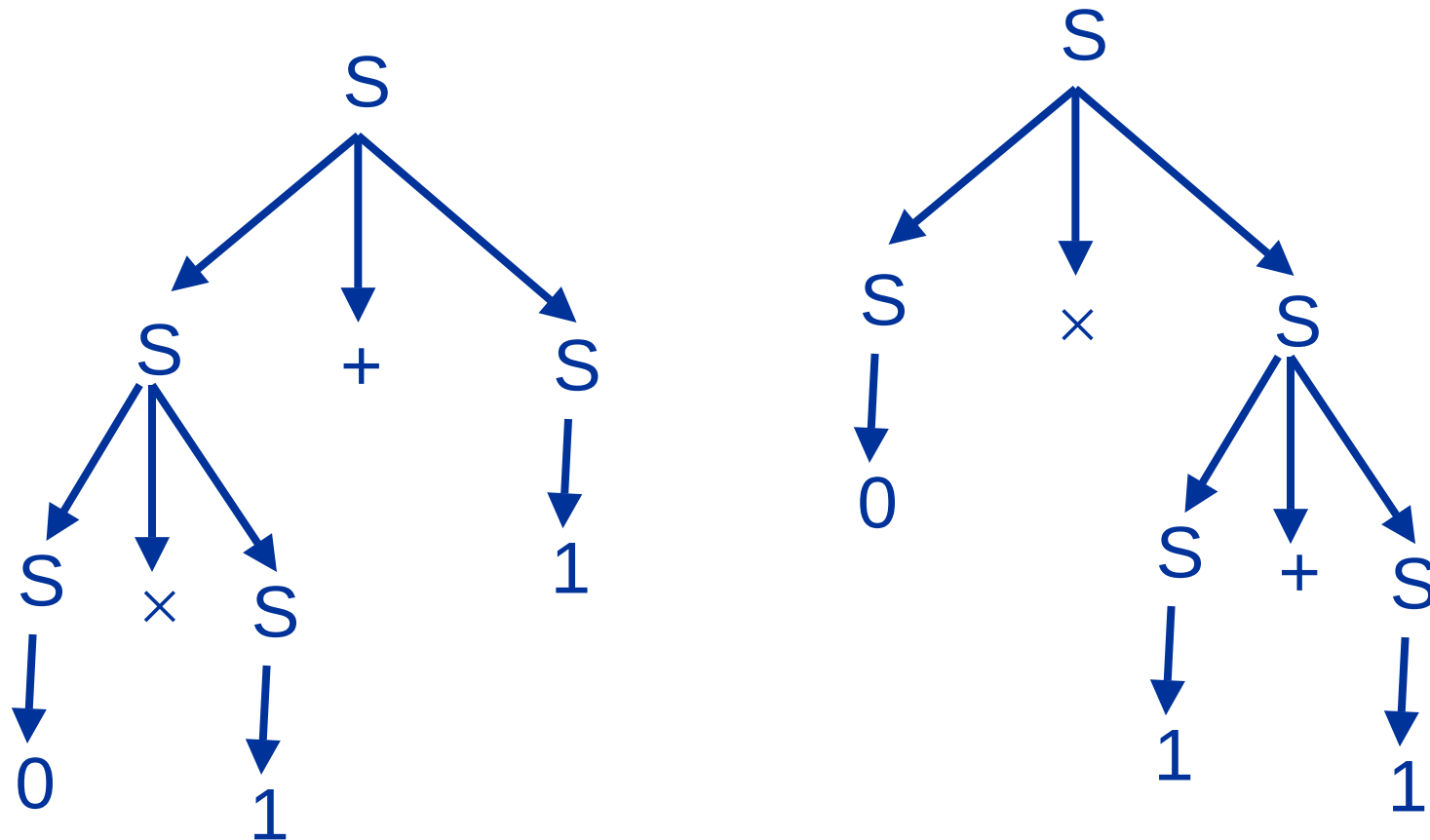
Typical example: rule $S \rightarrow 0 \mid 1 \mid S+S \mid S \times S$

$S \Rightarrow S+S \Rightarrow S \times S+S \Rightarrow 0 \times S+S \Rightarrow 0 \times 1+S \Rightarrow 0 \times 1+1$
versus

$S \Rightarrow S \times S \Rightarrow 0 \times S \Rightarrow 0 \times S+S \Rightarrow 0 \times 1+S \Rightarrow 0 \times 1+1$

Ambiguity and Parse Trees

The ambiguity of $0 \times 1 + 1$ is shown by the two different parse trees:



More on Ambiguity

The two different derivations:

$$S \Rightarrow S+S \Rightarrow 0+S \Rightarrow 0+1$$

and

$$S \Rightarrow S+S \Rightarrow S+1 \Rightarrow 0+1$$

do *not* constitute an ambiguous string 0+1
(they will have the same parse tree)

Languages that can only be generated by
ambiguous grammars are “inherently ambiguous”

Context-Free Languages

Any language that can be generated by a context free grammar is a context-free language (CFL).

The CFL $\{ 0^n 1^n \mid n \geq 0 \}$ shows us that certain CFLs are nonregular languages.

Q1: Are all regular languages context free?

Q2: Which languages are outside the class CFL?

“Chomsky Normal Form”

A context-free grammar $G = (V, \Sigma, R, S)$ is in Chomsky normal form if every rule is of the form

$$A \rightarrow BC$$

or $A \rightarrow x$

with variables $A \in V$ and $B, C \in V \setminus \{S\}$, and $x \in \Sigma$

For the start variable S we also allow the rule

$$S \rightarrow \varepsilon$$

Advantage: Grammars in this form are far easier to analyze.

Theorem 2.9

Every context-free language can be described by a grammar in Chomsky normal form.

Outline of Proof:

We rewrite every CFG in Chomsky normal form.

We do this by replacing, one-by-one, every rule that is not 'Chomsky'.

We have to take care of: Starting Symbol, ε symbol, all other violating rules.

Proof of Theorem 2.9

Given a context-free grammar $G = (V, \Sigma, R, S)$, rewrite it to Chomsky Normal Form by

- 1) New start symbol S_0 (and add rule $S_0 \rightarrow S$)
- 2) Remove $A \rightarrow \varepsilon$ rules (*from the tail*):
before: $B \rightarrow xAy$ and $A \rightarrow \varepsilon$, after: $B \rightarrow xAy \mid xy$
- 3) Remove unit rules $A \rightarrow B$ (*by the head*): “ $A \rightarrow B$ ” and “ $B \rightarrow xCy$ ”, becomes “ $A \rightarrow xCy$ ” and “ $B \rightarrow xCy$ ”
- 4) Shorten all rules to two: before: “ $A \rightarrow B_1 B_2 \dots B_k$ ”,
after: $A \rightarrow B_1 A_1$, $A_1 \rightarrow B_2 A_2, \dots, A_{k-2} \rightarrow B_{k-1} B_k$
- 5) Replace ill-placed terminals “ a ” by T_a with $T_a \rightarrow a$

Proof of Theorem 2.9

Given a context-free grammar $G = (V, \Sigma, R, S)$, rewrite it to Chomsky Normal Form by

- 1) New start symbol S_0 (and add rule $S_0 \rightarrow S$)
- 2) Remove $A \rightarrow \varepsilon$ rules (*from the tail*):
before: $B \rightarrow xAy$ and $A \rightarrow \varepsilon$, after: $B \rightarrow xAy \mid xy$
- 3) Remove unit rules $A \rightarrow B$ (*by the head*): “ $A \rightarrow B$ ” and “ $B \rightarrow xCy$ ”, becomes “ $A \rightarrow xCy$ ” and “ $B \rightarrow xCy$ ”
- 4) Shorten all rules to two: before: “ $A \rightarrow B_1 B_2 \dots B_k$ ”,
after: $A \rightarrow B_1 A_1$, $A_1 \rightarrow B_2 A_2, \dots, A_{k-2} \rightarrow B_{k-1} B_k$
- 5) Replace ill-placed terminals “ a ” by T_a with $T_a \rightarrow a$

Careful Removing of Rules

Do not introduce new rules that you removed earlier.

Example: $A \rightarrow A$ simply disappears

When removing $A \rightarrow \varepsilon$ rules, insert *all* new replacements:

$B \rightarrow AaA$ becomes $B \rightarrow AaA \mid aA \mid Aa \mid a$

Example of Chomsky NF

Initial grammar: $S \rightarrow aSb \mid \varepsilon$

In Chomsky normal form:

$$S_0 \rightarrow \varepsilon \mid T_a T_b \mid T_a X$$

$$X \rightarrow S T_b$$

$$S \rightarrow T_a T_b \mid T_a X$$

$$T_a \rightarrow a$$

$$T_b \rightarrow b$$

$RL \subseteq CFL$

Every regular language can be expressed by a context-free grammar.

Proof Idea:

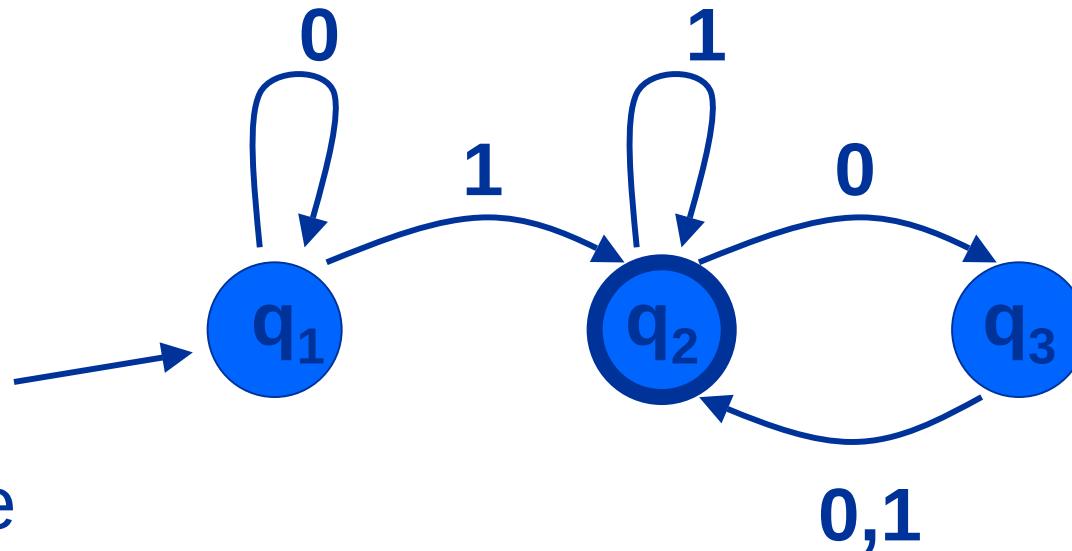
Given a DFA $M = (Q, \Sigma, \delta, q_0, F)$, we construct a corresponding CF grammar $G_M = (V, \Sigma, R, S)$ with $V = Q$ and $S = q_0$

Rules of G_M :

$$\begin{array}{ll} q_i \rightarrow x \delta(q_i, x) & \text{for all } q_i \in V \text{ and all } x \in \Sigma \\ q_i \rightarrow \varepsilon & \text{for all } q_i \in F \end{array}$$

Example $RL \subseteq CFL$

The DFA



leads to the
context-free grammar

$G_M = (Q, \Sigma, R, q_1)$ with the rules

$q_1 \rightarrow 0q_1 \mid 1q_2$

$q_2 \rightarrow 0q_3 \mid 1q_2 \mid \varepsilon$

$q_3 \rightarrow 0q_2 \mid 1q_2$

Picture Thus Far

